IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: David Meiri	Art Unit	: 2157
Serial No.	: 09/768,323	Examiner	: Barbara M. Burgess
Filed	: January 24, 2001	Conf. No.	: 3938
Title	: INTER-PROCESSOR MESSAGING		

MAIL STOP APPEAL BRIEF-PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

(1) Real Party in Interest

The real party in interest is EMC Corporation, a corporation of Massachusetts having a place of business at 35 Parkwood Drive, Hopkinton, MA, as evidenced by an assignment executed January 19, 2001 and recorded at the U.S. Patent Office on January 24, 2001 at Reel 011487, Frame 0177.

(2) Related Appeals and Interferences

There are no related appeals or interference.

(3) Status of Claims

Claims 1-9 stand rejected, are pending and on appeal. Of these, claim 1 is independent.

(4) Status of Amendments

All amendments have been entered.

(5) Summary of Claimed Subject Matter

CLAIM 1

"A method for posting a message on a message list accessible to a plurality of processors" is described in connection with FIG. 4, on page 10 of the specification. The message list **34** is described on page 7, lines 26-27 and shown in FIG. 3. The message list

resides in a shared memory **12** accessible to processor **28**, as recited on page 7, lines 17-22. Specifically, "selecting a new-message slot" is described in connection with step **62** in FIG. 4 on page 10, lines 3-8.

The limitation of "placing said message in said new-message slot" is described in connection with step **64** of FIG. 4 on page 10, lines 20-21. The limitation of "modifying said new-message slot to specify an intended recipient of said message" in described in connection with step **66** of FIG. 4, page 10, lines 21-23. That the intended recipient is selected from a plurality of processors is apparent from the description of the destination mask **52** on page 9, lines 5-8.

(6) Grounds of Rejection to be Reviewed on Appeal

Claims 1-9 stand rejected under 35 USC §103(a) as being rendered obvious by the combination of *Chou*, et al., U.S. Patent No. 5,905,897 and *Kingsbury,* et al. U.S. Patent Publ. No. 2003/0061395.

(7) Argument

Section 103 rejection of claims 1-9 based on *Chou* and *Kingsbury*

Applicant's invention

Applicant's invention concerns management of a message list. The message list includes messages that are intended for different processors.

The cited art fails to disclose a message list in which the messages are intended for different processors.

Overview of *Kingsbury*

Kingsbury discloses nodes **12**, **14**, **16**, **18** linked to each other as shown in FIG. 1:

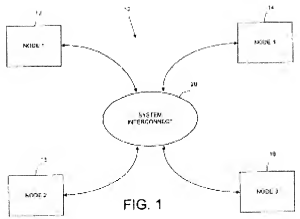


FIG. 1

A typical node can include multiple processors **24**, memory **26**, and other components typical of a computer in data communication with other computers on a network:[1]

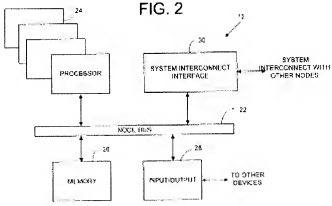


FIG. 2

In *Kingsbury*, a process[2] can receive a message from another process. In order to receive a message, a process must be associated with a mailbox data structure.[3] A mailbox data structure associated with a particular process is shown in FIG. 5:

[1] *Kingsbury*, para. 27.

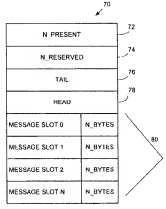


FIG. 5

To send a message to a receiving process, a sending process first finds the mailbox data structure for the receiving process. Then, the sending process stores the message in one of the slots in receiving process' mailbox. [4] In particular, *Kingsbury* states that

> **[0038]** Once a mailbox data structure for a process is allocated and initialized, a sending process can proceed to send messages to that process via its mailbox. In the case of

In describing software for creating the mailbox data structure, *Kingsbury* further states that a mailbox data structure "serves as a receiving area for messages being sent to a *specific* recipient process."[5]

> process. Block **42** represents the code for creating and initializing "mailbox" data structures (each mailbox of which serves as a receiving area for messages being sent to a specific recipient process). Block **42** also represents the

[2] *Kingsbury* distinguishes between *processors* and *processes*. Although processors participate in the execution of processes, the two are not the same.

[3] *Kingsbury*, para. 31.

[4] *Kingsbury*, para. 31.

[5] *Kingsbury*, para. 29, [emphasis supplied].

A mailbox for a process is therefore very much like a mailbox for an apartment. Just as an apartment's mailbox is associated with a particular apartment, a process' mailbox **70** is associated with a particular process. A mailbox for an apartment would only have messages for that apartment, not for other apartments. In the same way, a mailbox for a process would only have messages for that process, not for other processes.

***Kingsbury* lacks a message list with messages for different recipients**

Claim 1 requires a message list that includes messages having different intended recipients. Specifically, claim 1 recites

> receiving, from one of the plurality of processors, a message to be posted on said message list, said message having an intended recipient selected from said plurality of processors having access to said message list, **wherein said message list includes messages having different intended recipients**;

According to the Examiner, message slots **80** in *Kingsbury*'s mailbox data structure **70** include "messages having different intended recipients."

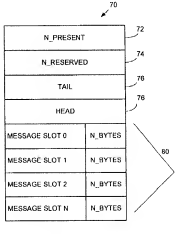


FIG. 5

In fact, the four message slots shown in FIG. 5 hold messages for the same intended recipient. All messages stored in the mailbox shown in FIG. 5 are intended for the process that owns that mailbox.

As discussed above, each process in *Kingsbury* has its own mailbox data structure **70**. To send a message to a receiving process, a sending process places the message in the receiving process' mailbox. One of ordinary skill in the art would have realized immediately that placing messages for a second process in a mailbox owned by a first process would result in the second process never receiving the message. One of ordinary skill in the art would have understood that such an error would be comparable to placing a letter intended for one apartment into a mailbox for a different apartment.

Thus, the Examiner's assertion:

> Therefore, Kingsbury undoubtedly discloses a single message list contains messages for all processors and a message list that includes messages having different intended recipients [6]

is manifestly inconsistent what one of ordinary skill in the art would have understood from reading *Kingsbury*.

The Examiner has pointed out that in *Kingsbury*, processors in one node can access memory in another node. This suggests that, using its mailbox, a process can receive messages from many different processes, even those running on different nodes. It does not mean that a process' mailbox has messages intended for different processes as well.

By way of analogy, a mail carrier who inserts letters into the various mailboxes of an apartment building certainly has "access" to each mailbox. But it is still the case that each mailbox has letters intended for a particular apartment.

***Kingsbury* fails to teach inter-processor communication**

Kingsbury discloses a system for communication between processes, not between processors. For example, *Kingsbury*'s abstract states:

---

[6] *Final Office Action*, page 11.

A lock-free mechanism is provided for successfully passing messages between processes even if a process is preempted while sending or receiving a message. Messages are communicated between processes using a mailbox data structure stored in memory shared by the processes, without the use of locks or other mutual exclusion entities that would otherwise limit concurrent delivery and receipt of messages placed in the data structure. The data structure in the

Although the words "process" and "processor" share a common root, they mean completely different things to one of ordinary skill in the art. For example, a single processor can spawn multiple processes that communicate with each other.

Accordingly, it is Applicant's position that *Kingsbury* does not disclose "receiving" a message "from one of [a] plurality of processors."

The Examiner has pointed out that *Kingsbury* discloses nodes that have more than one processor.

Applicant fails to see the relevance. The fact that a node has two or more processors does not necessarily mean that the processors are communicating with each other, let alone doing so by posting messages on a "message list" that is "accessible to" those processors, as claim 1 requires.

Overview of *Chou*

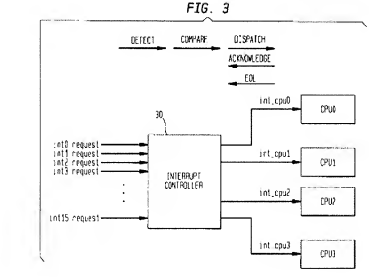
Chou discloses a way to handle interrupt requests from peripheral devices such as a keyboard, mouse, monitor, printer, or disk drive:[7]

FIG. 3 shows a multiprocessor system in which the present invention may be implemented, and illustrates the basic steps of interrupt processing. The system includes an interrupt controller 30 which is configured to detect and latch in sixteen different interrupt requests int0 through int15. These interrupt requests may be received from peripheral devices such as a keyboard, mouse, monitor, printer and disk drive as well as other elements of a computer system.

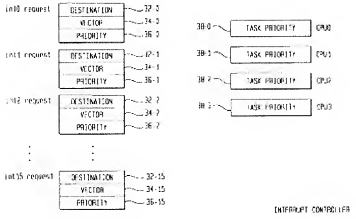
[7] *Chou*, col. 4, lines 40-47.

These interrupt requests are dispatched to various CPUs by an interrupt controller **30**, as shown in FIG. 3:

FIG. 3

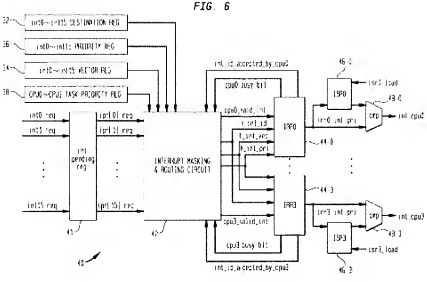


According to FIG. 4, each interrupt request includes a destination register **32-0**, a vector register **34-0**, and a priority register **36-0**:



The interrupt requests are stored in an interrupt pending register **41** within the interrupt controller **30**:

FIG. 6



***Chou* fails to teach message list accessible to processors**

The Examiner reasons as follows:

1. Each "interrupt request" is a "message."

2. The "interrupt pending register" **42**, in which the "interrupt requests" are stored, is a "message list."

3. Since the interrupt controller **30** connects to multiple CPU's (see FIG. 3), those CPUs can access register **42**.

The Examiner's logic is flawed. The mere fact that a controller can send a message to a CPU does not mean that the register that stores the message within the controller is accessible to that CPU.

By way of counter-example, a first computer can send an email message to a second computer. But this certainly does not mean that the second computer has access to the memory that the first computer used to store the email message.

Accordingly, *Chou* fails to disclose any "message list" that is "accessible to a plurality of processors." Therefore *Chou* cannot disclose or suggest "selecting a new-message slot from said message list accessible to said plurality of processors," as required by claim 1.

***Chou* fails to teach modifying message slot to specify intended recipients**

Claim 1 recites the limitation of

> "modifying said new-message slot to specify said intended recipient of said message,"

Chou discloses an interrupt request having a destination **32-0**, as shown in FIG. 4:

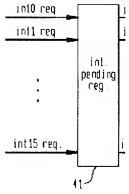
According to the Examiner: the above interrupt request corresponds to claim 1's "message," and destination **32-0** identifies "an intended recipient of said message."

> According to Chou, the interrupt request (message) has an intended recipient which is identified and selected by the destination processor and interrupt controller (column 3, lines 25-35).

Also according to the Examiner, *Chou*'s pending register **41** corresponds to claim 1's "message slot:"

- Selecting a new-message slot (column 3, lines 3-5, 11-16, column 4, lines 42-45, 66-67. The interrupt controller has pending registers in which interrupt requests are stored).

As shown in FIG. 6, the pending register **41** receives interrupt requests:



The Examiner overlooks the fact that according to claim 1, it is the "message slot" itself that is modified, and not the message. Since the "slot" corresponds to the "pending register," *Chou* would have to disclose modifying the pending register **41** "to specify said intended recipient of said message." *Chou* fails to disclose this. Instead, *Chou* discloses modifying the interrupt request itself (specifically, destination **32-0**), not the pending register **41**.

Motivation to combine is flawed

As a motivation to combine the references, the Examiner states:

> Therefore, one of ordinary skill in the art at the time the invention was made would have found it obvious to implement or incorporate Kingsbury message list in Chou's method enabling nodes to pass messages to each other [8]

According to KSR, "rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness"[9]

The proposed motivation to combine is a mere conclusory statement. The Examiner does not advance any reason for why one of ordinary skill in the art would have considered doing so.

---

[8] *Final Office Action*, page 4.

[9] *KSR Intern. Co. v. Teleflex Inc.*, 127 S.Ct. 1727, 1741 (2007).

Chou discloses directing an interrupt request to an appropriate one of several processors. These interrupt requests come from peripheral devices, like a keyboard, mouse, or disk drive.

Kingsbury discloses allowing a process (not to be confused with "processor") to communicate with another process by providing a mailbox for each process.

There is no plausible basis for why one of ordinary skill in the art would combine such dissimilar teachings. Nor has the Examiner provided any plausible basis, beyond the conclusory assertion that it is somehow obvious to do so.

Applicant recognizes that common sense of those having ordinary skill in the art is a basis for combining references. However, there is no common sense that would suggest combining a method for processing of interrupt requests from such devices as a mouse or keyboard with the sending messages from one process to another process.

The proposed motivation to combine the references is thus insufficient to support a prima facie case of obviousness.

SECTION 103 REJECTION OF CLAIM 2

Claim 2 recites the additional limitation of:

> "inserting said new-message slot into said message list, said message list including a first existing-message slot having a pointer to a second existing-message slot".

The Office suggests that the foregoing limitation is disclosed at *Kingsbury* paragraphs 13-14 and 34.

***Kingsbury* fails to teach inserting a message slot into a message list**

The claim limitation recites the inserting a message slot into the message list. This is not the same as inserting a message into a message slot.

These two operations are fundamentally different. Inserting a slot in a list changes the number of slots in the list, and hence changes the underlying data structure. In

contrast, inserting a message into an existing slot does not change the number of slots in the list, and hence does not change the underlying message structure.

By way of analogy, in the example referring to apartment mailboxes, it is one thing to drop a letter into an existing mailbox. It is something altogether different to actually make a new mailbox.

The cited paragraphs (paragraphs 13-14) do not disclose inserting a message slot into message list.

Paragraph 14 refers to inserting messages into slots, as shown below:

> **[0014]** In one aspect of the invention, the data structure may include a set of message slots storing messages sent to the data structure and a set of indicators manipulatable by an atomic operation for inserting messages into and removing messages from the message slots. The indicators may be a 10

But as noted above, the claim requires inserting a message slot into the message list, not inserting a message into a message slot.

Paragraph 13 describes an "availability indicator" that indicates whether a slot is available or not. According to paragraph 46, the availability indicator is N_RESERVED, the value of which represents how many slots are filled or filling. An "availability indicator" simply indicates whether any slots in the mailbox are available. It does not point to any slot. Therefore, it cannot be regarded as "a pointer to a second existing-message slot."[11]

Moreover, the "availability indicator" is not associated with "a first existing message slot" or any message slot. The "availability indicator" is part of the mailbox header.

---

[10] *Kingsbury*, para. 14.

[11] *Kingsbury* indicates that some of the state variables in a mailbox can be "pointers or other objects that relate to the indicated data". Such pointers presumably point to a location at which the data of interest is stored, not a pointer to a slot.

***Kingsbury* fails to disclose a message slot with pointers**

Claim 2 requires

> "a first existing-message slot having a pointer to a second existing-message slot"

The only plausible pointers described in *Kingsbury* are TAIL **76** and HEAD **78**. Both of these point to particular message slots **80**. The Examiner appears to recognize this on page 12 of the Final Action:

> (k) Kingsbury teaches the use of indicators that provide information on the status of data structure. The indicators may also be pointers. The tail index is a location indicator that identifies the location of an available message slot in the mailbox structure (message slot) (paragraphs [0034, 0047]). [12]

However, the Examiner then makes the error of assuming that a pointer to a message slot must itself be part of another message slot:

> Therfore, Kingsbury undoubtedly discloses an availability indicator is a pointer from one message slot to another. [13]

In fact, the foregoing statement is factually incorrect. According to FIG. 5, messages are plced in message slots 0-N. The remaining locations simply describe the state of the mailbox. Therefore, neither TAIL nor HEAD are part of any message slot. Accordingly, neither TAIL nor HEAD amount to pointers from an "existing message slot having a pointer to a second existing message slot."

SECTION 103 REJECTION OF CLAIM 3

Claim 3 recites the additional limitation of

> "setting a first pointer on said new-message slot to point to said first existing-message slot and a second pointer on said new-message slot to point to said second existing message-slot"

---

[12] *Final Office Action*, page 12.

[13] *Final Office Action*, page 12.

The Examiner suggests that setting *Kingsbury*'s "availability indicators" corresponds to the claim's step of "setting a first pointer."

Kingsbury's "availability indicator" (N_RESERVED) indicates the number of slots either filled or currently filling. It does not point to any particular slot.

In addition, like TAIL and HEAD, N_RESERVED is not part of any message slot. Accordingly, setting a value for N_RESERVED does not amount to "setting a first pointer on said new-message slot" because N_RESERVED is not part of any message slot.

SECTION 103 REJECTION OF CLAIM 4

Claim 4 requires

> "setting said pointer associated with said first existing-message slot to point to said new-message slot"

The Examiner suggests that *Kingsbury*'s "availability indicators" correspond in some way to this claim limitation.

Kingsbury's "availability indicator" (N_RESERVED) "indicates the number of slots either filled or currently filling." It therefore does not point to any particular slot.

In addition, the "availability indicator" is not part of any slot. Accordingly, setting its value does not amount to "setting said pointer associated with said first existing-message slot to point to said new-message slot" because the "availability indicator" is not part of any message slot.

SECTION 103 REJECTION OF CLAIM 5

Claim 5 recites the additional limitation of

> "modifying a destination mask associated with said new-message slot, said destination mask including information specifying all intended recipients of said message"

The Examiner has already suggested, in rejecting claim 4, that *Kingsbury*'s "availability indicators" are pointers to a new message slot. In rejecting claim 8, the

Examiner suggested that the "availability indicators" are "attention masks". Now, in rejecting claim 5, the Examiner suggests that the "availability indicators" are also a "destination mask."

> However, Kingsbury teaches availability indicators that indicate if a message slot is available to receiving messages, if a message is present in a message slot, and that a message is no longer present in the message slot. Indicators also show the number of slots filled or currently filling as well as a full mailbox (paragraphs [0013-0014, 0034]). [14]

Kingsbury's "availability indicator" (N_RESERVED) simply "indicates the number of slots either filled or currently filling." Thus, N_RESERVED is simply a number. Knowing how many slots are filled or filling has nothing to do with specifying intended recipients of a message.

The Examiner further suggests that a node number, port number, and internal and external names for a mailbox somehow function as a "destination mask."

> (i) According to Kingsbury, each mailbox (message slot) has an external and internal name both having addresses and consisting of node number and port number (destination masks) (paragraphs [0036-0037]). Therefore, Kingsbury discloses destination masks that include information specifying intended recipients of a particular message. [15]

However, the node number and port number are not associated with a new-message slot. In addition, the node number and port number are simply numbers that identify one node and one port on that node. They do not specify "all intended recipients" of any message. Nor is there any indication that these node numbers and port numbers are ever modified, as would be required by the limitation "modifying a destination mask."

---

[14] *Final Office Action*, page 6.

[15] *Final Office Action*, page 12.

Similarly, the internal and external names for a mailbox just identify a mailbox. They do not specify "all intended recipients of" a message. Nor are they ever modified as the claim requires.

SECTION 103 REJECTION OF CLAIM 6

Claim 6 recites the additional limitation of

> selecting, from a plurality of constituent data-elements of said destination mask, each of said constituent data-elements corresponding to one of said processors from said plurality of processors, a selected data-element corresponding to a selected processor; and
>
> modifying said selected data-element to indicate that said selected processor is an intended recipient."

The Examiner suggests that this claim limitation is somehow connected with *Kingsbury*'s "availability indicators":

> However, Kingsbury teaches availability indicators that indicate if a message slot is available to receiving messages, if a message is present in a message slot, and that a message is no longer present in the message slot. Indicators also show the number of slots filled or currently filling as well as a full mailbox (paragraphs [0013-0014, 0034]). [16]

Kingsbury's "availability indicator" (N_RESERVED) "indicates the number of slots either filled or currently filling." This plainly has nothing to do with selecting data elements from a destination mask. In fact, since N_RESERVED is simply a single number that specifies how many slots are filled or currently filling, it would not really have "a plurality of constituent data elements" each of which corresponds "to one of said processors." After all, there is no particular relationship between the number of slots in the mailbox data structure and the number of nodes in the system.

[16] *Final Office Action*, page 7.

Further, there is no indication that N_RESERVED, or any of the other variables in the mailbox header, are ever modified "to indicate that said selected processor is an intended recipient" of a message.

SECTION 103 REJECTION OF CLAIM 7

Claim 7 recites the additional limitation of

> "updating a message directory to indicate the presence of said new-message slot in said message list, said message directory being accessible to said plurality of processors."

The Examiner suggests that this claim limitation is disclosed in paragraphs 34 and 45.

Paragraph 45 describes a memory fence operation that ensures a correct value of "N_BYTES" for a particular message slot. There is no modification of anything that would indicate the presence of a new message slot.

Paragraph 34 describes the four variables in the header of a mailbox data structure. These four variables do not indicate how many message slots are in the mailbox data structure.

For example, in FIG. 5, there are five message slots. While the values of N_PRESENT, N_RESERVED, TAIL and HEAD might change in the course of operation, as these slots fill with messages, and as those messages are processed, there would always be the same five message slots. Accordingly, modifying any of the variables N_PRESENT, N_RESERVED, TAIL and HEAD, whether singly or in combination, would never "indicate the presence of said new-message slot in said message list".

SECTION 103 REJECTION OF CLAIM 8

Claim 8 recites the additional limitation of

> "updating an attention mask containing information indicative of which processors from said plurality of processors are intended recipients of messages contained in said message list"

Kingsbury's "availability indicators," which played the role of "destination masks" in the rejection of claim 5, and the role of a "pointer" in rejecting claim 4, now find themselves recruited to play yet another role: that of an "attention mask" in claim 8:

> However, Kingsbury teaches availability indicators that indicate if a message slot is available to receiving messages, if a message is present in a message slot, and that a message is no longer present in the message slot. Indicators also show the number of slots filled or currently filling as well as a full mailbox (paragraphs [0013-0014, 0034])

As has already been discussed in connection with claims 5 and 8, the "availability indicators" do not play any role in specifying "intended recipients of messages contained in" the mailbox.

SECTION 103 REJECTION OF CLAIM 9

Claim 9 recites the further limitation of

> "selecting from a plurality of constituent data-elements of said attention mask, each of said constituent data-elements corresponding to one of said processors from said plurality of processors, a selected data-element corresponding to a selected processor; and
>
> modifying said selected data-element to indicate existence of a new message for which said selected processor is an intended recipient."

In rejecting claim 9, the Examiner calls upon the "availability indicators" one last time:

> However, Kingsbury teaches availability indicators that indicate if a message slot is available to receiving messages, if a message is present in a message slot, and that a message is no longer present in the message slot. Indicators also show the number of slots filled or currently filling as well as a full mailbox (paragraphs [0013-0014, 0034]). [17]

[17] *Final Office Action*, page 8.

As has already been discussed above, the "availability indicators" do not play any role in indicating "the existence of a new message for which said selected processor is an intended recipient."

Respectfully submitted,

Date: April 23 2010

Faustino A. Lichauco
Reg. No. 41,942

Customer No. 69713
Occhiuti Rohlicek & Tsao LLP
10 Fawcett Street
Cambridge, MA 02138
Telephone: (617) 500-2533
Facsimile: (617) 500-2499
139903.doc

(8) Claims Appendix

1. (**Previously Presented**) A method for using a computer to assist a particular data storage machine in posting a message on a message list stored in a memory, said message list being accessible to a plurality of processors, said method comprising:

receiving, from one of the plurality of processors, a message to be posted on said message list, said message having an intended recipient selected from said plurality of processors having access to said message list, wherein said message list includes messages having different intended recipients;

selecting a new-message slot from said message list accessible to said plurality of processors;

placing said message to be posted on said message list in said new-message slot; and

modifying said new-message slot to specify said intended recipient of said message, said intended recipient being selected from said plurality of processors having access to said message list.

2. (**Original**) The method of claim **1** further comprising inserting said new-message slot into said message list, said message list including a first existing-message slot having a pointer to a second existing-message slot.

3. (**Original**) The method of claim **2** wherein inserting said new-message slot into said message list comprises setting a first pointer on said new-message slot to point to said first existing-message slot and a second pointer on said new-message slot to point to said second existing message-slot.

4. (**Original**) The method of claim **3** wherein inserting said new-message slot into said message list further comprises setting said pointer associated with said first existing-message slot to point to said new-message slot.

5. (**Original**) The method of claim **1** wherein modifying said new-message slot to specify an intended recipient comprises modifying a destination mask associated with said new-message slot, said destination mask including information specifying all intended recipients of said message.

6. (**Original**) The method of claim **5** wherein modifying said destination mask comprises:

selecting, from a plurality of constituent data-elements of said destination mask, each of said constituent data-elements corresponding to one of said processors from said plurality of processors, a selected data-element corresponding to a selected processor; and

modifying said selected data-element to indicate that said selected processor is an intended recipient.

7. (**Original**) The method of claim **1** further comprising updating a message directory to indicate the presence of said new-message slot in said message list, said message directory being accessible to said plurality of processors.

8. (**Original**) The method of claim **7** wherein updating said message directory comprises updating an attention mask containing information indicative of which processors from said plurality of processors are intended recipients of messages contained in said message list.

9. (**Previously Presented**) The method of claim **8** wherein updating said attention mask comprises:

selecting from a plurality of constituent data-elements of said attention mask, each of said constituent data-elements corresponding to one of said processors from said plurality of processors, a selected data-element corresponding to a selected processor; and

modifying said selected data-element to indicate existence of a new message for which said selected processor is an intended recipient.

(9) Evidence Appendix

None

10. Related Proceedings Appendix

None